



廣東財經大學

GUANGDONG UNIVERSITY OF FINANCE & ECONOMICS

课程设计

项目名称 网络爬虫抓取三万本豆瓣图书

班级与班级代码 12 级计算机科学与技术实验班

专 业: 计算机科学与技术

学 号: 12251003102

姓 名: 陈锦瀚

日 期: 2015 年 12 月 10 日

姓名 陈锦瀚

课程设计成绩 _____

评语：

指导教师（签名） _____

2015 年 12 月 日

目录

一、 前言.....	2
二、 分析与设计.....	2
1、入口分析.....	2
2、抓取设计.....	5
3、爬取成果.....	6
三、 系统实现.....	7
1、系统环境.....	7
2、结果展示.....	7
3、 程序详解.....	10
抓取主程序.....	11
文本提取程序.....	22
图片抓取程序.....	26
抓取辅助工具.....	27
文件处理工具.....	29
数据库辅助工具.....	31
代理 IP 存放工具.....	34
代理文件格式.....	35
数据库设计.....	35
四、 总结.....	36
参考.....	37

网络爬虫抓取豆瓣三万本图书

一、前言

随着互联网的发展,越来越多的人凭借浏览器就可以和五湖四海的人进行交流和分享。由于因特网上有庞大的资源,但是这些资源常常是收费或者难以得到。比如百度音乐上的很多音乐只有VIP会员才可以进行下载,而某些网站只有缴费才能得到API接口的授权进行数据的获取。

突然有一天,我发现我想看小说或者看电影,但我不知道哪些比较好看,于是我上了豆瓣浏览,其中有图书、电影等的信息以及各种人的评论,我发现不同的人对不同的书籍或电影有不同的见解,我很享受这个过程。可是我想整理一个书单打印下来,里面有书籍简介什么的,可是我一张张网页的点击以及复制粘贴的速度简直哔了勾。如何解决这个问题呢?

HTTP 协议是一个基于请求与响应模式的、无状态的、应用层的协议,常基于 TCP 的连接方式,大多数的网址都通过此协议进行信息的传送,比如网址 URL 一般是 <http://hunterhug.github.io:8080/index.html>,请参考百度百科:HTTP 协议。通过 URL,远程服务器会发送 HTML, XML, JSON 等格式的文件,文本信息就储存于此。

动态语言 Python 是一个写网络爬虫较好的语言,通过模拟浏览器,通过 URL 链接可以将远程的 HTML 文件抓取储存在本地,并且可以对其中内容进行文本分析,比如提取下级链接进行多层爬取,提取关键内容存入本地数据库或 EXCEL 文件等。

通过分析,我发现豆瓣图书频道有大量的书籍可以进行爬取,爬取到的包括图书封面,图书简介,图书评分,图书评价甚至读者信息等。爬取到的这些数据的用处很多,至少满足了我随时随地了解这些书和其他人的看法。

此课程设计的目的是抓取豆瓣所大部分图书信息,包括这些图书的大部分热门评论,当然包括一些用户信息,虽然学校网络限制和反爬虫 IP 被封,但爬取几万本书是没问题的。在本文的最后会给出爬取的图书数目,总评论数。

二、分析与设计

此课程设计的逻辑如下:

分析网站入口——>抓取网页——>存储网页——>提取数据

先分析网站 URL 入口,一层层观察,找出爬虫可以爬取的路径,然后分析网页有那些内容可以进行爬取,最后分析爬取的网页需要如何存储,如何解决某些已经爬过的链接。当然设计是分析后的实现,会进行 Python 关键代码的讲解。

因为爬取的量特别巨大,所以提取数据是最后才做的,一开始要尽量把所有网页爬取到本地,再进行后续的文本挖掘等。

1、入口分析

进入豆瓣官网 <http://www.douban.com/>,点击读书。进入读书频道可以发现许多入口,但是我们关心的是大类,寻找很久后发现热门标签涵盖大部分书的入口,点击进入。如下图。

图一：豆瓣首页



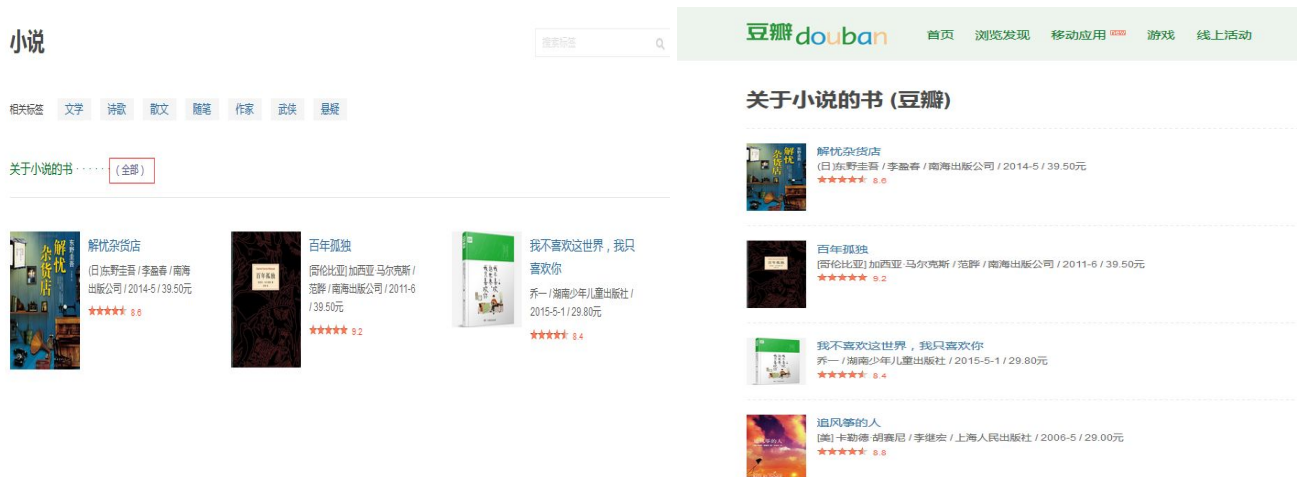
图二：读书频道和热门标签页



可以发现图书便签有六大类，一百多个标签。通过浏览器按 F12 进去开发者模式，进行链接观察可以发现标签链接为 <http://www.douban.com/tag/小说/?focus=book>，将其中小小说关键字替换为其他标签名即可。

点击 <http://www.douban.com/tag/小说/?focus=book> 进入该标签后出现以下图面，点击（全部）进入。可以发现一张图书列表，此时 URL 为 <http://www.douban.com/tag/小说/book>

图三：小说便签首页 小说图书列表页



进入浏览器开发模式，如下图所示，可以发现图书以 15 篇一页进行分页，URL 拼接后格式为：<http://www.douban.com/tag//book?start=15>

图四：图书列表分页观察



点击图书链接 http://book.douban.com/subject/25862578/?from=tag_all 可进入图书页，在该页上有图书简介，图书评论，可以进行抓取，如下图。

图五：图书页

关于小说的书 (豆瓣)

解忧杂货店
(日) 东野圭吾 / 李盈春 / 南海出版公司 / 2014-5 / 39.50元
★★★★★ 8.6

百年孤独
(哥伦比亚) 加西亚·马尔克斯 / 范晔 / 南海出版公司 / 2011-6 / 39.50元
★★★★★ 9.2

解忧杂货店

作者: (日) 东野圭吾
出版社: 南海出版公司
原作名: ナミヤ雑貨店の奇蹟
译者: 李盈春
出版年: 2014-5
页数: 291
定价: 39.50元
装帧: 精装
丛书: 新经典文库·东野圭吾作品
ISBN: 9787544270878

豆瓣评分: 8.6 (70632人评价)
5星: 49.3%
4星: 36.5%
3星: 11.2%
2星: 0.9%
1星: 0.2%

想读 在读 读过 评价: ☆☆☆☆☆
写笔记 写书评 加入购书单 添加到豆列 分享到

内容简介

现代人内心流失的东西，这家杂货店能帮你找回——
僻静的街道旁有一家杂货店，只要写下烦恼放进进投信的信箱，第二天就会在店后的牛奶箱里得到回答。
因男友身患绝症，年轻女孩静子在爱情与梦想间徘徊；克郎为了音乐梦想离家漂泊，却在现实中寸步难行；
少年浩介面临家庭巨变，挣扎在亲情与未来的迷茫中……
他们将困惑写成信投进杂货店，随即奇妙的事情竟不断发生。
生命中的一次偶然交会，将如何演绎出截然不同的人生？
如今回顾写作过程，我发现自己始终在思考一个问题：站在人生的岔路口，人究竟应该怎么做？我希望读者能在掩卷时喃喃自语：我从未读过这样的小说。——东野圭吾

作者简介

东野圭吾
日本著名作家。
1985年，《放学后》获第31届江户川乱步奖，开始专职写作；
1999年，《秘密》获第52届日本推理作家协会奖；
2005年出版的《嫌疑人X的献身》史无前例地同时获得第134届直木奖、第6届本格推理小说大奖，以及年
度三大推理小说排行榜第1名；
2008年，《流星之绊》获第43届新风奖；
2009年出版的《新参者》获两大推理小说排行榜年度第1名；
2012年，《解忧杂货店》获第7届中央公论文艺奖。
2014年，《祈りの幕が下りる時》（暂译《祈祷落幕时》）获第48届吉川英治文学奖。

目录

第一章 回答在牛奶箱里
第二章 深夜的口琴声
第三章 在思域车上等到天亮
第四章 听着披头士默祷
第五章 来自天上的祈祷

书评

热门评论 最新评论
我来评论这本书

呐，所谓羁绊就是甜蜜的负担啊
李小秀 (女人的勇气常被误认为是疯狂) ★★★★★
说来也是好笑，最近的我又开始重新追看《火影忍者》，即使中间隔了长长的十年时光，那些当初打动了我的东西，如今重温依然会让我鼻头泛酸。以至于昨晚我边看边和田妹说：“如果以后我们有孩子的话，我从小就给她看火影，让她远离国产垃圾动画的精神污染。因为火影把人与人之间的羁绊、人生在世需要肩负的责任都诠释的那么好，又不带一丝说教的意味。对了，还有这本书也可以一起看看。”我边说边晃了晃手上刚读完的《解忧杂货店》……
(118回应)

2014-06-13 14:41 1224/1305有用

技术贴：浪矢杂货店完整时间轴及人物手册
丁小叮 (好女孩上天堂，坏女孩走四方) ★★★★★
给身边很多人推荐了这本书，最不像东野圭吾的解忧杂货店。没有他引人入胜的悬疑和推理，看似凌乱实则缜密，每个人物都充满了矛盾和羁绊，最终又得到救赎。很多朋友反映此书人物众多，且时空错乱，搞不清神会觉得莫名其妙。为大家拥有更好的阅读体验，本人义不容辞制作了本技术贴，帮你理清时间顺序和人物关系。其实，这也是东野圭吾此书的奇妙之处，如有不当之处，欢迎大家多多交流和指正。【转载请注明出处】1960年浪矢雄治的老伴因……
(120回应)

以上只是简单的说明，通过网站的观察，可以发现爬取入口如下：

- 1、一级入口：<http://book.douban.com/tag/>
标签列表页，图书根据便签进行分类，提取所有标签，方便进入二级入口进行抓取。
- 2、二级入口：<http://www.douban.com/tag/小说/book>
标签页，根据一级入口抓取到的便签，如小说，文学等
- 3、三级入口：<http://book.douban.com/subject/25862578/>
三级入口：<http://book.douban.com/subject/25862578/reviews>
标签下单本书的详细页以及对应的评论列表页，方便抓取图书信息并提取对应的具体评论，包括评论内容，评论用户信息等。
- 4、四级入口：<http://book.douban.com/review/6700731/>
四级入口：<http://img4.douban.com/lpic/s27284878.jpg>
一个有完整内容的评论，还有图书封面

2、抓取设计

网页抓取，根据具体的语言进行 HTTP 请求头部的伪装，模拟浏览器访问远程网页。虽然可以边抓取网页边提取文本，但鉴于网络原因及爬取数量过大可能导致 IP 被封，先将其完整抓取保存在本地。

由于爬取过程中可能网络中断，程序重新运行时，爬虫应该自动判断哪个网页已经被抓取到。自动判断可以采用文件命名方法，每张网页每次爬取之前都自动生成一个固定文件名，判断文件夹下是否有该文件，没有的话进行爬取并储存，有则跳过。

由于二级入口需要根据一级入口的文本提取等，故无法一次性连贯爬取，所以抓取和提取的步骤是随时变动的，需采用敏捷开发模型，每个程序运行的反馈信息为下一个程序服务。

本文的难点是反爬虫，远程网站可能会识别出是机器人在爬取，故 HTTP 需要伪装头部，模拟浏览器，由于爬取速度过快，此时会出现 IP 被封，直接产生服务器拒绝访问 403 错误，为了降低速度，每次爬取一张页面爬虫休眠一秒钟，此时仍会出现 IP 被封。可以在浏览器客户端登录后复制 HTTP 头部 cookie 内容进行伪装，模拟登录，此时也会出现 403 错误，但从浏览器端打开会出现验证码页面，解决方法是爬虫休眠时间更长，模拟人类点击下一个链接的间隔时长。最后实在不行，程序可改由代理 IP 进行爬取，如何代理请见系统设计，代理大多数需要收费。

图六：爬虫被封图

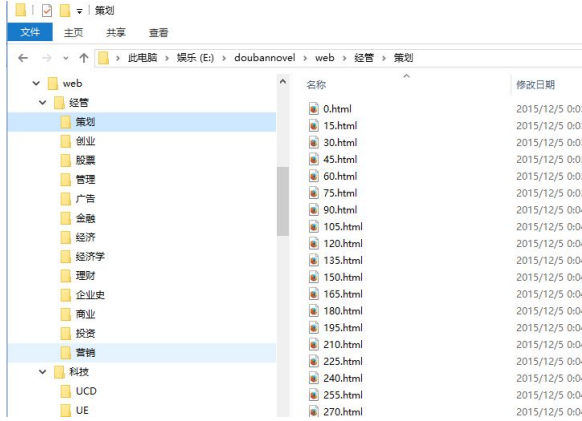
```
准备抓取: http://www.douban.com/tag/%E8%80%83%E5%8F%A4/book?start=270
标签: 考古
时间暂停: 0.19989953421966788
已经抓取: http://www.douban.com/tag/%E8%80%83%E5%8F%A4/book?start=270
标签: 考古
准备抓取: http://www.douban.com/tag/%E8%80%83%E5%8F%A4/book?start=285
标签: 考古
时间暂停: 0.19932308187048875
Traceback (most recent call last):
  File "E:/doubannovel/catch.py", line 148, in <module>
    catchbooklist(0,0.2,'lock3')
  File "E:/doubannovel/catch.py", line 85, in catchbooklist
HTTP Error 403: Forbidden
IP被封
webcontent = getHtml(site).encode('utf-8') # 爬取
File "E:/doubannovel/tool/gethtml.py", line 30, in getHtml
  html_bytes = urllib.request.urlopen(url).read()
File "C:\Python34\lib\urllib\request.py", line 161, in urlopen
  return opener.open(url, data, timeout)
```



对于网页如何存储在本地，存储的层次结构如何也是个问题，一个文件夹放一万个文件明显不行，于是决定按大分类，大分类下标签类的形式进行存储。对于文本提取时考虑到便利性，有些存放于 Excel 文件，大多数存放于数据库。

图七：使用代理和文件存放位置

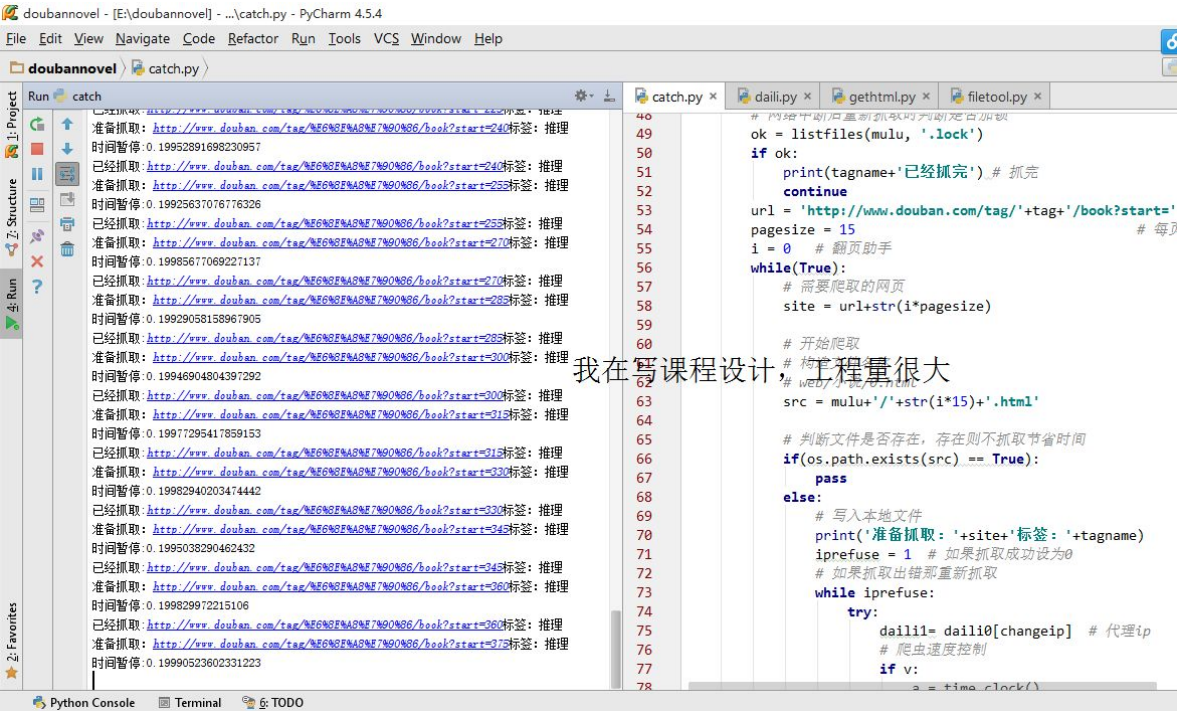
设计已经抓完
政治已经抓完
社会已经抓完
建筑已经抓完
宗教已经抓完
电影已经抓完
数学已经抓完
准备抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=312>标签: 政治学
代理: 181.111.192.146:8080
<urlopen error [WinError 10060] 由于连接方在一段时间内没有正确答复或连接的主机没有反应, 连接尝试失败。>
更换代理: 60.216.20.31:3128
已经抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=312>标签: 政治学
准备抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=320>标签: 政治学
代理: 60.216.20.31:3128
已经抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=320>标签: 政治学
准备抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=328>标签: 政治学
代理: 60.216.20.31:3128
已经抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=328>标签: 政治学
准备抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=336>标签: 政治学
代理: 60.216.20.31:3128
已经抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=336>标签: 政治学
准备抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=344>标签: 政治学
代理: 60.216.20.31:3128
已经抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=344>标签: 政治学
准备抓取: <http://www.douban.com/tag/%E6%94%B7%E6%82%B2%E5%A4%A6/book?start=352>标签: 政治学
代理: 60.216.20.31:3128
HTTP Error 500: Internal Server Error
更换代理: 183.63.129.91:3128
代理: 183.63.129.91:3128



3、爬取成果

爬取的成果应该是所有的原始网页，包括图书列表页，图书详情页，图书评论列表页，图书评论详情页，图书封面以及中途过程中文本提取得到的各种 Excel 文件。爬取得到的大量原始网页可以在后继进行文本挖掘时提取并保存进行数据库，数据库设计请见系统实现，由于篇幅和时间有限，数据库部分内容可能不会讲到。下图是爬取过程之一。

图八：爬取网页过程



三、系统实现

1、系统环境

机器：内存够大，CPU 够快的现代 PC 机

软件：Python3.4 及必备的库包，MySQL 版本>=5.6

网络：较通畅的宽带

软件下载参考：<http://www.cnblogs.com/nima>

2、结果展示

将图书插入数据库

```
Run catch
'外国文学', '文学')
查询语句: select * from `book` where `bookno`='26393693'
执行语句成功:INSERT INTO `book` (`bookname`,`bookurl`,`bookimg`,`bookinfo`,`bookstar`,`bookno`) VALUES
('达·芬奇笔记(大开本精装版)', 'http://book.douban.com/subject/26393693/?from=tag_all', 'http://img3.douban
.com/lpic/s28076324.jpg', '[意] 列奥纳多·达·芬奇 著 / [美] H·安娜·苏 编 / 刘勇 / 湖南科学技术出版社 / 2015-5 /
238元', '9.0', '26393693')
查询语句: select * from `booktag` where `bookno`='26393693' and `booktag`='外国文学' and `bookkind`='文学'
执行语句成功:INSERT INTO `booktag` (`bookname`,`bookno`,`booktag`,`bookkind`) VALUES ('达·芬奇笔记(大开本精装版)',
'26393693', '外国文学', '文学')

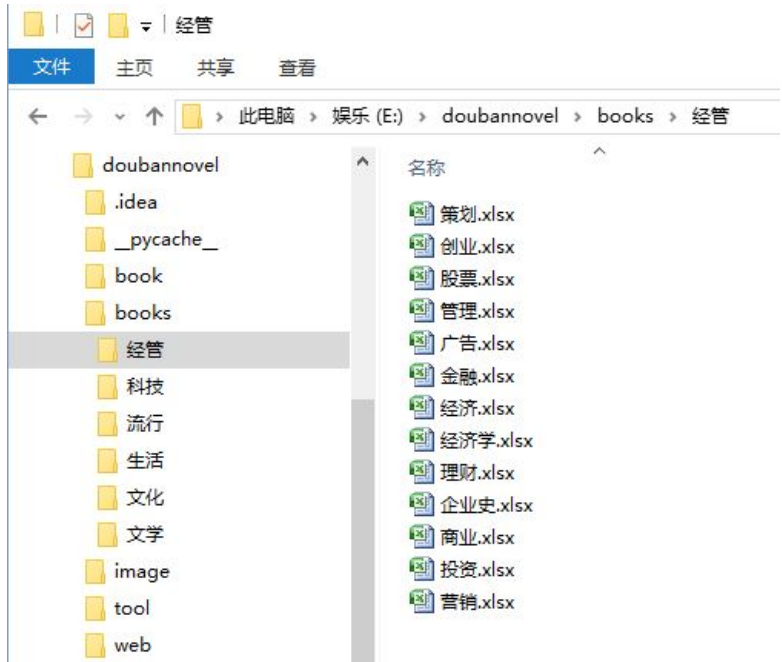
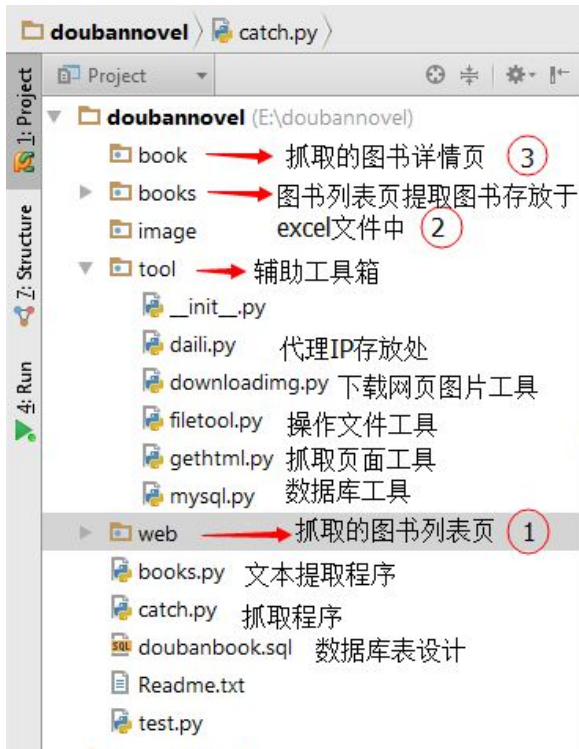
查询语句: select * from `book` where `bookno`='3057556'
执行语句成功:INSERT INTO `book` (`bookname`,`bookurl`,`bookimg`,`bookinfo`,`bookstar`,`bookno`) VALUES ('狂热分子',
'http://book.douban.com/subject/3057556/?from=tag_all', 'http://img3.doubanio.com/lpic/s28036829.jpg', '[美]
埃里克·霍弗 / 梁永安 / 广西师范大学出版社 / 2011-6 / 34.00元', '9.1', '3057556')
查询语句: select * from `booktag` where `bookno`='3057556' and `booktag`='外国文学' and `bookkind`='文学'
执行语句成功:INSERT INTO `booktag` (`bookname`,`bookno`,`booktag`,`bookkind`) VALUES ('狂热分子', '3057556', '外国文学',
'文学')
```

```
Run catch
C:\Python34\python.exe E:/doubannovel/catch.py
读取数据成功!
读取数据成功!
查询语句: select * from `book` where `bookno`='25862578'
解忧杂货店: http://book.douban.com/subject/25862578/?from=tag_all已经存在
查询语句: select * from `booktag` where `bookno`='25862578' and `booktag`='小说' and `bookkind`='文学'

查询语句: select * from `book` where `bookno`='6082808'
百年孤独: http://book.douban.com/subject/6082808/?from=tag_all已经存在
查询语句: select * from `booktag` where `bookno`='6082808' and `booktag`='小说' and `bookkind`='文学'

查询语句: select * from `book` where `bookno`='26414020'
我不喜欢这世界, 我只喜欢你: http://book.douban.com/subject/26414020/?from=tag_all已经存在
查询语句: select * from `booktag` where `bookno`='26414020' and `booktag`='小说' and `bookkind`='文学'
```

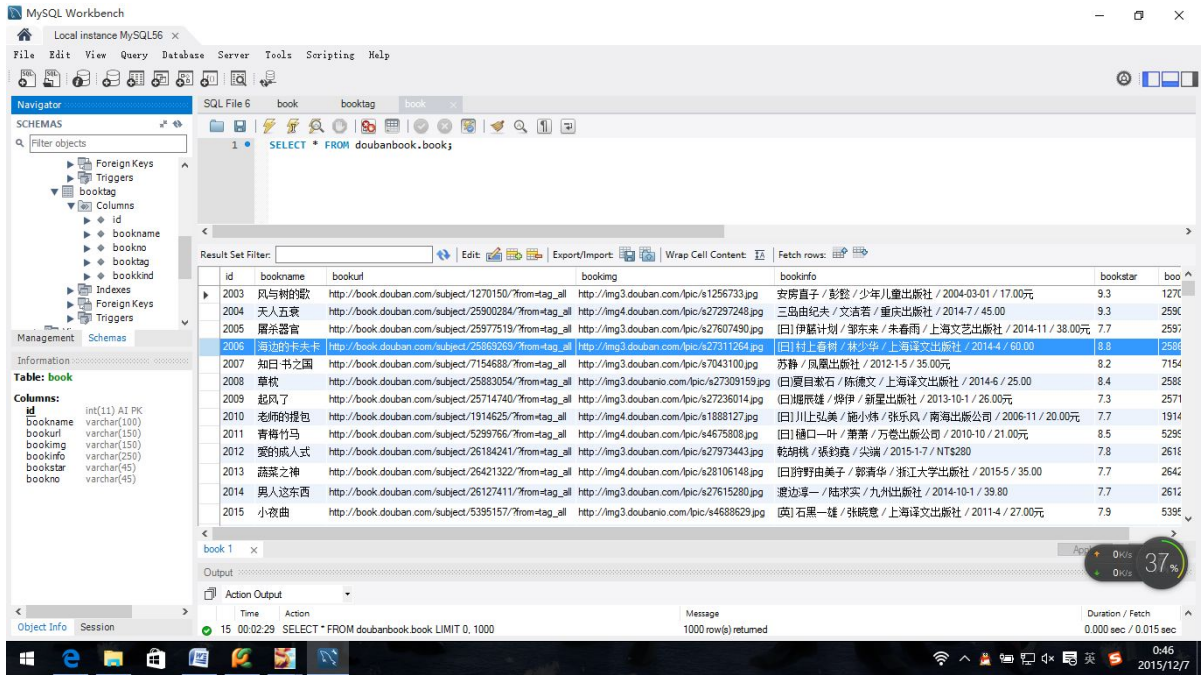
项目结构及图书列表提取中间产物



图书列表提取中间产物 excel

序号	书籍名	URL入口	图片地址	出版信息	评价星数
1	解忧杂货店	http://book.douban.com/subject/25862578/?from=tahtp://img4.douban.com/lpic/s27284878.jpg	(日) 东野圭吾 / 李盈春 / 南海出版公司 / 8.6		
2	百年孤独	http://book.douban.com/subject/6082808/?from=taghttp://img3.douban.com/lpic/s6384944.jpg	[哥伦比亚] 加西亚·马尔克斯 / 范晔 / 9.2		
3	我不喜欢这世界	http://book.douban.com/subject/26414020/?from=tahtp://img3.douban.com/lpic/s28096601.jpg	芥一 / 湖南少年儿童出版社 / 2015-5-1 / 8.4		
4	追风筝的人	http://book.douban.com/subject/1770782/?from=taghttp://img3.douban.com/lpic/s1727290.jpg	[美] 卡勒德·胡赛尼 / 李继宏 / 上海人民 / 8.8		
5	火星救援	http://book.douban.com/subject/26586492/?from=tahtp://img3.douban.com/lpic/s28315660.jpg	[美] 安迪·威尔 / 陈韵 / 译林出版社 / 9.0		
6	且将生活一切	http://book.douban.com/subject/26648238/?from=tahtp://img3.douban.com/lpic/s28323823.jpg	李国峰 / 中国华侨出版社 / 2015-11-1 / 8.4		
7	三体	http://book.douban.com/subject/2567698/?from=taghttp://img4.douban.com/lpic/s2768378.jpg	刘慈欣 / 重庆出版社 / 2008-1 / 23.00 / 8.8		
8	你的一切好	http://book.douban.com/subject/26652086/?from=tahtp://img3.douban.com/lpic/s28328655.jpg	杨西溪 / 北京联合出版公司 / 2015-11 / 9.1		
9	小王子 (纯美	http://book.douban.com/subject/26647054/?from=tahtp://img3.douban.com/lpic/s28322662.jpg	[法] 安东尼·德·圣-埃克苏佩里 / 梅思繁 / 9.3		
10	无声告白	http://book.douban.com/subject/26382433/?from=tahtp://img3.douban.com/lpic/s28109182.jpg	[美] 伍绮诗 / 孙璐 / 江苏凤凰文艺出版社 / 8.2		
11	白夜行	http://book.douban.com/subject/3259440/?from=taghttp://img3.douban.com/lpic/s4610502.jpg	[日] 东野圭吾 / 刘姿君 / 南海出版公司 / 9.1		
12	小王子	http://book.douban.com/subject/1084336/?from=taghttp://img3.douban.com/lpic/s1237549.jpg	[法] 圣埃克苏佩里 / 马振聘 / 人民文学 / 9.0		
13	岛上书店	http://book.douban.com/subject/26340138/?from=tahtp://img3.douban.com/lpic/s28049685.jpg	[美] 加布瑞埃拉·泽文 / 孙仲旭 / 李天 / 9.9		
14	活着	http://book.douban.com/subject/4913064/?from=taghttp://img3.douban.com/lpic/s27279654.jpg	余华 / 作家出版社 / 2012-8-1 / 20.00 / 9.3		
15	生活与命运	http://book.douban.com/subject/26394018/?from=tahtp://img4.douban.com/lpic/s28265238.jpg	[俄罗斯] 瓦西里·格罗斯曼 / 力冈 / 广西 / 8.7		
16	穆斯林的葬礼	http://book.douban.com/subject/1082334/?from=taghttp://img3.douban.com/lpic/s1790771.jpg	霍达 / 北京十月文艺出版社 / 1988-12-1 / 8.3		
17	昨日的世界	http://book.douban.com/subject/25868351/?from=taghttp://img4.douban.com/lpic/s1121598.jpg	[奥] 斯蒂芬·茨威格 / 舒昌善 / 孙龙生 / 9.2		
18	耶路撒冷三千年	http://book.douban.com/subject/26609056/?from=tahtp://img3.douban.com/lpic/s28293203.jpg	[英] 西蒙·蒙蒂菲奥里 / 张倩红 / 马丹静 / 8.1		
19	年少荒唐	http://book.douban.com/subject/4117922/?from=taghttp://img3.douban.com/lpic/s4055190.jpg	朱往 / 江苏凤凰文艺出版社 / 2015-9 / 27.9		
20	嫌疑人的献	http://book.douban.com/subject/26628984/?from=tahtp://img3.douban.com/lpic/s28327229.jpg	(日) 东野圭吾 / 刘子倩 / 南海出版公司 / 9.0		
21	小王子	http://book.douban.com/subject/25796120/?from=taghttp://img3.douban.com/lpic/s28077170.jpg	圣埃克苏佩里 / 尹建莉 / 尹建莉 / 新蕾 / 9.4		
22	小径分岔的花	http://book.douban.com/subject/26650970/?from=tahtp://img3.douban.com/lpic/s28340239.jpg	[阿根廷] 博尔赫斯 / 王永年 / 上海译文 / 9.1		
23	古董局中局	http://book.douban.com/subject/1088581/?from=taghttp://img4.douban.com/lpic/s1075516.jpg	马伯庸 / 北京联合出版公司 / 2015-12 / 7.5		
24	浮生六记	http://book.douban.com/subject/1948901/?from=taghttp://img3.douban.com/lpic/s4442295.jpg	(清) 沈复 / 人民文学出版社 / 1999-1 / 8.9		
25	盗墓笔记	http://book.douban.com/subject/26576518/?from=tahtp://img3.douban.com/lpic/s28265889.jpg	南派三叔 / 中国友谊出版公司 / 2007-1 / 8.4		
26	我的失落	http://book.douban.com/subject/26576518/?from=tahtp://img3.douban.com/lpic/s28265889.jpg	东野圭吾 / 代珂 / 南海出版公司 / 2015-7.3		

图书数据库信息



插入数据库结束

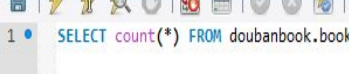
合并图书列表进数据库总共运行时间：13034.746 秒

```
Process finished with exit code 0
```



1 • `SELECT bookkind, count(*) FROM doubanbook.booktag group by bookkind`

Result Set Filter: | Export: | Wrap Cell Content:

bookkind	count(*)
文化	16829
文学	11792
流行	13519
生活	10341
科技	5275
经管	6200



1 • `SELECT count(*) FROM doubanbook.book`

Result Set Filter:  Export:  V

count(*)
34702

图书详情数据库数据









id	bookname	bookno	bookinfo
1000	建筑的七重奏	1074780	作者 (美)约翰·罗斯金 出版社 山东画报出版社 副标题 The Seven Lamps of Architecture 原书名 (英)约翰·罗斯金 (John Ruskin) 译者 曹方 译者附注 出版社 山东画报出版社 译者 曹方 译者附注 原书名 Nature's Architecture 译者 曹方 译者附注 原书名 2006-07-01 页数 336 定价 38.00 装帧 平装 ISBN 9787547302557
998	自然权利与历史	1074728	作者 (美)詹姆斯·马奇 出版社 译林出版社 副标题 都会政治史与理论 原书名 Malinche's Anarchy 译者 张华建 译者附注 出版社 译林出版社 译者 张华建 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
996	让子弹飞	1073827	作者 姜文 译者 姜文 译者附注 出版社 译林出版社 副标题 姜文电影研究 原书名 姜文 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
994	普普艺术	1073566	作者 (美)安迪·沃霍尔 出版社 江苏人民出版社 副标题 1983-1984 美国艺术界事件 原书名 Andy Warhol's Pop Art 译者 张华建 译者附注 出版社 江苏人民出版社 译者 张华建 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
992	欧洲简史	1073362	作者 (法)亨利·特罗亚 出版社 商务印书馆 副标题 为了便于人们认识和掌握欧洲历史而写的简明历史 原书名 亨利·特罗亚 译者 王元化 译者附注 出版社 商务印书馆 译者 王元化 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
990	欧洲文明的危机	1073354	作者 (法)亨利·特罗亚 出版社 商务印书馆 副标题 为了便于人们认识和掌握欧洲历史而写的简明历史 原书名 亨利·特罗亚 译者 王元化 译者附注 出版社 商务印书馆 译者 王元化 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
988	意大利简史	1073358	作者 (法)亨利·特罗亚 出版社 商务印书馆 副标题 为了便于人们认识和掌握欧洲历史而写的简明历史 原书名 亨利·特罗亚 译者 王元化 译者附注 出版社 商务印书馆 译者 王元化 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
986	俄罗斯简史	1072936	作者 (法)亨利·特罗亚 出版社 商务印书馆 副标题 为了便于人们认识和掌握欧洲历史而写的简明历史 原书名 亨利·特罗亚 译者 王元化 译者附注 出版社 商务印书馆 译者 王元化 译者附注 原书名 2012-07-01 页数 256 定价 38.00 装帧 平装 ISBN 9787544635562
984	这个历史不该忘记	1072757	作者 曹方 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
982	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
980	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
978	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
976	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
974	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
972	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
970	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
968	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
966	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
964	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
962	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
960	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
958	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
956	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
954	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
952	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
950	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
948	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
946	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
944	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
942	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
940	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
938	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
936	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 331 定价 38.00 装帧 平装 ISBN 9787544635562
934	世界历史大事年表	1072723	作者 姜文 出版社 译林出版社 副标题 中国历史上 出版社 2016-04-01 页数 33

以上为部分截图，通过统计，豆瓣图书有 30000 多本。

关于图书评论页的数量过大，如果需要抓取，可参考抓取图书列表页。

3、程序详解

代码运行，按照以下顺序。

 step1.py	2015/12/14 13:51	JetBrains PyChar...	1 KB
 step2.py	2015/12/14 14:20	JetBrains PyChar...	1 KB
 step3.py	2015/12/14 14:21	JetBrains PyChar...	1 KB
 step4.py	2015/12/14 14:28	JetBrains PyChar...	1 KB
 step5.py	2015/12/14 14:29	JetBrains PyChar...	1 KB
 step6.py	2015/12/14 14:30	JetBrains PyChar...	1 KB
 step7.py	2015/12/16 13:00	JetBrains PyChar...	1 KB
 step8.py	2016/1/2 13:58	JetBrains PyChar...	1 KB

以下为各步骤代码，具体代码太多。

```
# -*- coding:utf-8 -*-
import tool.mysql
# 新建数据库
result = tool.mysql.initdoubanbook()
print(result)

# -*- coding:utf-8 -*-
from tool.gethtml import getHtml
import bookdeal
# 抓取分类标签页
tag = getHtml('http://book.douban.com/tag/')
file = open('web/booktag.html', 'wb')
file.write(tag.encode())
file.close()

# 抓取列表页方便测试
tag1 = getHtml("http://www.douban.com/tag/%E5%B0%8F%E8%AF%B4/book")
file1 = open('web/books.html', 'wb')
file1.write(tag1.encode())
file1.close()

# 抓取图书页方便测试
tag3 = getHtml("http://book.douban.com/subject/25862578/?from=tag_all")
file2 = open('web/book.html', 'wb')
file2.write(tag3.encode())
file2.close()
print("成功")

# -*- coding:utf-8 -*-
import bookdeal
```

```

# 提取标签页到excel
bookdeal.testbooktag()

# -*- coding:utf-8 -*-
import catch
# 抓取各标签列表页
catch.catchbooklist(0,2,'lock3')

# -*- coding:utf-8 -*-
import catch
# 提取各标签列表页到excel
catch.dealbooklist()

# -*- coding:utf-8 -*-
import catch
# 合并各标签列表页 excel 到数据库
catch.mergeboolist()

# -*- coding:utf-8 -*-
import catch
# 抓取图书详情页
catch.catchbook(0,0,34800)#1900

# -*- coding:utf-8 -*-
import catch
# 处理提取图书详情页
catch.dealbook()

```

抓取主程序

catch.py

```

# -*- coding:utf-8 -*-
# http://book.douban.com/tag/
# http://www.douban.com/tag/小说/book?start=0 书列表 间隔15
# http://book.douban.com/subject/25862578/?from=tag_all 书信息
# http://book.douban.com/subject/6082808/reviews?score=&start=0 书评 间隔25
from tool.gethtml import getHtml,getBinaryHtml
import time
import os.path
from tool.filetool import listfiles,readexcel,writeexcel,validateTitle
import bookdeal

```



```

import urllib.error
import urllib.parse
import re
from tool.daili import daili
from tool.mysql import Mysql
from pymysql import escape_string
#web504=['1002582','1010668',, '10459781']#
web504=['1010668','1023322','10459781','1915375']
# 抓取书表: 第一步
def catchbooklist(requreip = 0, v=0, lockprefix= 'lock'):
    """
    输入参数为:
    是否使用代理, 默认否
    是否限制爬虫速度, 默认否, 时间为1 秒仿人工
    文件加锁后缀
    """
    # 进行计时
    start = time.clock()
    taglist = readexcel('web/booktag.xlsx') # 读取标签
    daili0 = daili() # 代理IP 数组
    changeip = 0 # 代理ip 下标
    # 循环对标签进行抓取
    for i in range(1,len(taglist)):
        kinds = taglist[i][0] # 大分类
        tagname = taglist[i][1] # 标签名
        tag = urllib.parse.quote(tagname) # url 中文转码
        mulu0 = 'web/'+kinds
        # 存在大分类文件夹则跳过
        if os.path.exists(mulu0):
            pass
        else: # 否则新建
            print('新建大分类: '+mulu0)
            os.makedirs(mulu0)

        mulu = mulu0+'/'+tagname
        # 存在标签文件夹则跳过
        if os.path.exists(mulu):
            pass
        else: # 否则新建方便网页存放
            print('新建标签文件夹'+mulu)
            os.makedirs(mulu)

        # 网络中断后重新抓取时判断是否加锁
        ok = listfiles(mulu, '.'+lockprefix)

```

```

if ok:
    print('类别: '+kinds+'----标签: '+tagname+'----已经抓完') # 抓完
    continue
url = 'http://www.douban.com/tag/'+tag+'/book?start=' # 基础网址
pagesize = 15 # 每页 15 本
i = 0 # 翻页助手
while(True):
    # 需要爬取的网页
    site = url+str(i*pagesize)

    # 开始爬取
    # 构造文件名称
    # web/小说/0.html
    src = mulu+'/' +str(i*15)+'.html'

    # 判断文件是否存在, 存在则不抓取节省时间
    if(os.path.exists(src) == True):
        pass
    else:
        # 写入本地文件
        print('准备抓取: '+site+'类别: '+kinds+'----标签: '+tagname)
        iprefuse = 1 # 如果抓取成功设为0
        # 如果抓取出错那重新抓取
        while iprefuse:
            try:
                daili1= daili0[changeip] # 代理ip
                # 爬虫速度控制
                if v:
                    a = time.clock()
                    time.sleep(v)
                    b = time.clock()
                    print('时间暂停:'+str(b-a))
                # 不需要代理
                if requireip==0:
                    webcontent = getHtml(site).encode('utf-8') # 爬取
                    # print(webcontent.decode('utf-8', 'ignore'))
                    notnull =
re.search(r'<dl>',webcontent.decode('utf-8', 'ignore')) # 匹配看是否抓取到末
页

                iprefuse = 0 # 抓完设置0
            else: # 需要代理
                print('代理: '+daili1)
                webcontent = getBinaryHtml(site, daili1)
                # print(webcontent.decode('utf-8', 'ignore'))

```

```

        notnull =
re.search(r'<dl>', webcontent.decode('utf-8', 'ignore'))
        print(notnull)
        iprefuse = 0
    except Exception as e:
        print(e)
        if requireip:
            changeip = changeip+1 # 更换ip 下标
            if changeip==len(daili0): # 到达ip 数组末循环再来
                changeip = 0
            print('更换代理: '+daili0[changeip])
        else:
            print("IP 被封")
            raise
        return
    # break

# 如果抓不到<dl>标签, 证明已经抓取完
if notnull:
    webfile = open(src, 'wb')
    webfile.write(webcontent)
    webfile.close()
    print("已经抓取:"+site+'类别: '+kinds+'----标签: '+tagname)
else:
    lock = open(src.replace('html',lockprefix),'w') # 加锁证明抓完
    # 日期:
    http://blog.csdn.net/caisini_vc/article/details/5619954
    finish = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
    lock.write('抓取完成时间: '+finish)
    print("抓取完毕: "+tagname)
    break

    i =i + 1 # 加页
# 计时
end = time.clock()
print("爬取总共运行时间 : %.03f 秒" %(end-start))

# 分析提取书表: 第二步
def dealbooklist():
    start = time.clock()
    putplace = 'books'
    # 判断存放位置是否存在
    if os.path.exists(putplace):
        pass
    else: # 否则新建

```

```

print('新建图书提取存放 excel 处: '+putplace)
os.makedirs(putplace)
taglist = readexcel('web/booktag.xlsx') # 读取标签列表
del taglist[0]
# 对于每个标签
for tag in taglist:

    # 图书按照标签存放于文件夹中
    mulu=putplace+'/'+tag[0]
    if os.path.exists(mulu):
        pass
    else:
        os.makedirs(mulu)

    excelpath = mulu+'/'+tag[1]+'.xlsx'
    # 存在处理过的excel 文件则跳过
    if os.path.exists(excelpath):
        print(excelpath+'已经存在')
        continue

    tagbooks = [] # 该标签所有书存放处
    path = 'web/'+tag[0]+'/'+tag[1] # 构造读取文件夹入口
    print('本地提取: '+path)
    # 查找目录下已经抓取的Html
    files = listfiles(path)
    # 遍历分析
    for i in files:
        file = path+'/'+i
        print('提取: '+file)
        content = open(file,'rb').read()
        book = bookdeal.manybook(content) # 提取图书列表
        for j in book: # 重新包装图书
            # print('提取: '+', '.join(j))
            tagbooks.append(j)

    # 将信息写入本地文件中
    booksattr=['书籍名','URL 入口','图片地址','出版信息','评价星数']
    tagbooks.insert(0,booksattr)
    writeexcel(excelpath,tagbooks)
    print('写入成功: '+excelpath)
end = time.clock()
print("提取图书列表总共运行时间 : %.03f 秒" %(end-start))

```

书表去重并写入数据库: 第三步

```

# 读取Excel, 判断是否重复, 先加入 book 表, 重复则往 booktag 表插入标签记录
def mergeboolist():
    start = time.clock()
    taglist = readexcel('web/booktag.xlsx') # 读取标签列表
    del taglist[0]
    database = Mysql(host="localhost", user="root", pwd="6833066",
db="doubanbook")
    for tag in taglist: # 遍历所有标签
        kind = tag[0] # 大类
        tagname = tag[1] # 标签
        excelpath = 'books/'+kind+'/'+tagname+'.xlsx' # 本地文件
        try:
            datas = readexcel(excelpath)
        except Exception as e:
            print(e)
            continue
        del datas[0] # 去掉标题
        #print(datas)
        # 提取图书插入数据库
        for data in datas:
            bookname = data[0].replace("'", "\'").replace('"', '\\"')
            bookurl = data[1].replace("'", "\'").replace('"', '\\"')
            bookimage = data[2].replace("'", "\'").replace('"', '\\"')
            bookno =
bookurl.split('/')[2].replace("'", "\'").replace('"', '\\"')
            try:
                bookinfo = data[3].replace("'", "\'").replace('"', '\\"')
            except:
                bookinfo = ''
            pass
            try:
                bookstar = data[4]
            except:
                bookstar = '0'
            pass
            # select * from `book` where `bookno`='dc'
            searchsql1 = "select * from `book` where `bookno`='"+bookno+"'"
            print(searchsql1)
            try:
                isexist1 = database.ExecQuery(searchsql1)
            except Exception as e:
                print(e)
                pass
            # 如果图书记录存在, 插Booktag 表

```



```

    if isexist1:
        print(bookname+':'+bookurl+'已经存在')

    else:
        insertbooksql = "INSERT INTO `book` (`bookname`, `bookurl`,
`booking`, `bookinfo`, `bookstar`, `bookno`) VALUES ('" \
                        "{bookname}', '{bookurl}', '{booking}', '{bookinfo}',
'{bookstar}', '{bookno}')"
        insert1 = insertbooksql.format(bookname=bookname,
bookurl=bookurl, booking=bookimage, bookinfo=bookinfo, bookstar=bookstar,
bookno=bookno)
        print(insert1)
        try:
            database.ExecNonQuery(insert1)
        except Exception as e:
            print(e)
        pass
        # 如果图书标签存在, 则不插入
        searchsql = "select * from `booktag` where `bookno`='{bookno}' and
`booktag`='{booktag}' and `bookkind`='{bookkind}'"
        searchsql2 =
searchsql.format(bookno=bookno, booktag=tagname, bookkind=kind)
        print(searchsql2)
        try:
            isexist2 = database.ExecQuery(searchsql2)
        except Exception as e:
            print(e)
        pass
        if isexist2.__len__()==0:
            inserttag = "INSERT INTO
`booktag` (`bookname`, `bookno`, `booktag`, `bookkind`) VALUES ('" \
                    "{bookname}', '{bookno}', '{booktag}', '{bookkind}')"
            insert2 = inserttag.format(bookname=bookname, bookno=bookno,
booktag=tagname, bookkind=kind)
            print(insert2)
            try:
                database.ExecNonQuery(insert2)
            except Exception as e:
                print(e)
            pass
        print('-'*100)
    print("插入数据库结束")
    end = time.clock()
    print("合并图书列表进数据库总共运行时间 : %.03f 秒" %(end-start))

```

```

# 抓取图书: 第四步
# 读取book表, 读取booktag表, 抓取图书网页拷贝多份到不同标签目录
def catchbook(requreip = 0, v=0, startbook=0):
    """
    输入参数为:
    是否使用代理, 默认否
    是否限制爬虫速度, 默认否, 时间为1秒仿人工
    startbook = 0 查询起始位置
    """
    # 进行计时
    start = time.clock()
    webe=[]
    selecttotal = 'select count(distinct bookno) from booktag'
    selectsql = 'SELECT bookname,bookkind,bookno FROM booktag group by bookno'
    database = Mysql(host="localhost", user="root", pwd="6833066",
db="doubanbook")
    total = database.ExecQuery(selecttotal) # 总记录
    total=int(total[0][0])
    daili0 = daili() # 代理IP数组
    dailino = 0
    changeip = 0 # 代理ip下标
    # 循环对分类进行抓取
    while startbook < total+100:
        selectsql1=selectsql+' limit '+str(startbook)+' ,100'
        taglist=database.ExecQuery(selectsql1)
        for i in range(0,len(taglist)):
            try:
                bookname = taglist[i][0]
                kinds = taglist[i][1] # 分类
                bookno = taglist[i][2] # 图书编号
                url = 'http://book.douban.com/subject/'+bookno # 抓取网址
                #http://book.douban.com/subject/25862578
            except:
                raise
            return
        mulu0 = 'book/'+kinds
        # 存在大分类文件夹则跳过
        if os.path.exists(mulu0):
            pass
        else: # 否则新建
            print('新建大分类: '+mulu0)
            os.makedirs(mulu0)

```

```

# 判断文件是否存在, 存在则不抓取节省时间
try:
    filename = mulu0+'/' +bookno+validateTitle(bookname)+'.html'
    if(os.path.exists(filename) == True):
        print(filename+'： 已经存在')
        continue
    elif bookno in web504:
        # 写入本地文件
        print('----'*5)
        print("504 错误,跳过: "+bookno)
        print('----'*5)
        continue
    else:
        #print("-"*50)
        print('准备抓取: '+url+'类别: '+kinds)
except:
    print(filename+"文件名异常")
    continue
iprefuse = 1 # 如果抓取成功设为0
# 如果抓取出错那重新抓取
while iprefuse:
    try:
        daili1= daili0[changeip] # 代理ip
        # 爬虫速度控制
        if v:
            a = time.clock()
            time.sleep(v)
            b = time.clock()
            print('时间暂停:'+str(b-a))
        # 不需要代理
        if requireip==0:
            webcontent = getHtml(url).encode('utf-8') # 爬取, 有时间限制,
            # 应对504 错误
            notnull = re.search(r'<div
class="top-nav-doubanapp">',webcontent.decode('utf-8','ignore'))
            if notnull:
                pass
            else:
                raise Exception("抓到的页面不是正确的页面"+filename)
            webfile = open(filename, 'wb')
            webfile.write(webcontent)
            webfile.close()
            print("已经抓取:"+url+'类别: '+kinds)
            iprefuse = 0 # 抓完设置0

```

```

else: # 需要代理
    print('代理: '+daili1)
    webcontent = getBinaryHtml(url, daili1)
    notnull = re.search(r'<div
class="top-nav-doubanapp">',webcontent.decode('utf-8','ignore'))
    if notnull:
        pass
    else:
        raise Exception("抓到的页面不是正确的页面"+filename)
    webfile = open(filename, 'wb')
    webfile.write(webcontent)
    webfile.close()
    print("已经抓取:"+url+'类别: '+kinds)
    iprefuse = 0
    dailino=dailino+1
    print('此次转换代理次数:'+str(dailino))
    if dailino>20:
        dailino=0
        requireip=0 # 代理100次后转为非代理
except urllib.error.URLError as e:
except Exception as e:
    print(url)
    if hasattr(e, 'code'):
        print('页面不存在或时间太长.')
        print('Error code:', e.code)
        if e.code==404:
            print('404 错误, 忽略')
            webe.append(bookno)
            break
    elif hasattr(e, 'reason'):
        print("无法到达主机.")
        print('Reason: ', e.reason)
    print(e)
    if requireip:
        changeip = changeip+1 # 更换ip 下标
        if changeip==len(daili0): # 到达ip 数组末循环再来
            changeip = 0
        print('更换代理: '+daili0[changeip])
        dailino=dailino+1
        print('此次转换代理次数:'+str(dailino))
        if dailino>20:
            dailino=0
            requireip=0 # 代理100次后转为非代理
    else:

```

```

        print("IP 被封或断网")
        requireip=1 # 转为代理
    print('已经抓了'+str(startbook+100)+'本')
    print()
    print()
    print()
    startbook=startbook+100
    if len(webe) > 20:
        print(webe)
        webep=open("book/book.txt",'a+')
        webep.write(','.join(webe)+'\n')
        webep.close()
        webe=[]
    else:
        pass
# 计时
end = time.clock()
print("爬取总共运行时间 : %.03f 秒" %(end-start))

# 提取图书: 第五步
# 扫描book 目录, 找出所有图书详情表进行提取, 插入数据库
def dealbook():
    rootdir='book'
    prefix='.html'
    database = Mysql(host="localhost", user="root", pwd="6833066",
db="doubanbook")
    insertbooksq1 = "INSERT INTO `bookdetial`
(`bookname`,`bookno`,`bookinfo`,`bookintro`,`authorintro`,`peoples`,`start
s`,`other`,`mulu`,`comments`) VALUES (" \
        "{0},{1},{2},{3},{4},{5},{6},{7},{8},{9})"
    for parent,dirnames,filenames in os.walk(rootdir):
        for filename in filenames:
            if filename.endswith(prefix) :
                path=str(parent)+'/'+filename
                print(path)
                content=open(path,'rb').read()
                try:
                    draw=bookdeal.onebook(content)
                except:
                    continue
            insert1 =
insertbooksq1.format(escape_string(draw[1]),draw[0],escape_string(draw[2]),

```



```

escape_string(draw[3]),\

escape_string(draw[4]),draw[5],escape_string(draw[6]),escape_string(draw[7]
),escape_string(draw[8]),escape_string(draw[9]))
    try:
        database.ExecNonQuery(insert1)
        os.rename(path,path+'lock1')
    except Exception as e:
        print(e)
        continue
    else:
        pass

if __name__=='__main__':
    #catchbooklist(0,2,'lock3')
    #dealbooklist()
    #mergeboolist()
    catchbook(0,3,18200)#1900

```

其他略，见附件代码

文本提取程序

bookdeal.py

```

# -*- coding:utf-8 -*-
from bs4 import BeautifulSoup
from tool.filetool import writeexcel
__author__ = 'hunterhug'
import json

def manybook(url_content):
    """
    抓取图书信息元组
    """
    books = []
    soup = BeautifulSoup(url_content, 'html.parser') # 开始解析

    booktable1 = soup.find_all("dl") # 找到所有图书所在标记

    # 循环遍历图书列表
    for book in booktable1:

```

```

simplebook = book

subsoup = BeautifulSoup(str(simplebook), 'html.parser') # 单本书进行解
析

# 图书封面:
# http://img4.doubanio.com/spic/s1237549.jpg
# http://img4.doubanio.com/lpic/s1237549.jpg
booksmallimg = subsoup.img['src']
imgtemp = booksmallimg.split('/')
imgtemp[len(imgtemp)-2] = 'lpic'
booklargeimg = '/'.join(imgtemp)

# 图书信息
booklink = subsoup.dd.a['href'] # 图书链接:
http://book.douban.com/subject/1084336/
bookname1 = subsoup.dd.a.string # 图书名称: 小王子
bookinfo = subsoup.div.string # 图书出版信息: [法] 圣埃克苏佩里 / 马振聘 /
人民文学出版社 / 2003-8 / 22.00 元
try:
    bookstar = subsoup.find('span', attrs={"class":
"rating_nums"}).string # 图书星级: 9.0
except:
    bookstar = ''
pass
bookinfo = bookinfo.strip(' \n')
books.append([bookname1, booklink, booklargeimg, bookinfo, bookstar])
# 返回图书列表
return books

def onebook(url_content):
    """
    抓取单本书

    """
    soup = BeautifulSoup(url_content, 'html.parser') # 开始解析
    bookno=soup.find('meta', attrs={'http-equiv': 'mobile-agent'})
    bookno=bookno['content'].split('subject/')[1].replace('/', '')
    bookname=soup.find('h1').text.replace('\n', '')
    #print(bookname)
    bookinfo = soup.find('div', attrs={"id": "info"}) # 出版信息
    bookp=soup.find('a', attrs={"class", "rating_people"})
    books=soup.findAll('span', attrs={"class", "rating_per"})

```

```

bookintro = soup.findAll('div',attrs={"class": "intro"}) # 书籍及作者介绍
bookalot = soup.findAll('div',attrs={"class": "subject_show block5"}) #
众书信息 可能不存在
bookamulu = soup.select('div[id*="dir"]')
bookhotcomment1 = soup.select('div#wt_1 div.ctsh div.tlst div.ilst a')
# 评论头像
bookhotcomment2 = soup.select('div#wt_1 div.ctsh div.tlst div.nlst h3 > a')
# 评论详情
bookhotcomment3 = soup.select('div#wt_1 div.ctsh div.tlst div.clst
span.starb') # 用户简介
try:
    bookinfo=bookinfo.text.replace(' \n','').replace('\n ','').replace('
','')
except:
    bookinfo=''
#print(bookinfo)
try:
    bookintro1=bookintro[0].findAll('p')
except:
    bookintro1=[]
try:
    bookintro2=bookintro[1].findAll('p')
except:
    bookintro2=[]
tro1=''
tro2=''
for i in bookintro1:
    tro1=tro1+i.text+'\n'
for i in bookintro2:
    tro2=tro2+i.text+'\n'
#print(tro2)
try:
    bookalot=bookalot[0].text.replace('\n','').replace(' ','')
except:
    bookalot=''
#print(bookalot)
bookp=bookp.text
try:
    bookamulu=bookamulu[0].text.replace(' ','')
    bookamulu=bookamulu[1].text.replace(' ','')
except:
    bookamulu=''
#print(bookhotcomment1[0])
#print(bookhotcomment2[0])

```

```

# print(bookhotcomment3[0])
peoples = []
for i in range(0, len(bookhotcomment1)):
    peoples.append('<br />'.join([str(bookhotcomment1[i]), str(bookhotcomment2[i]), str(bookhotcomment3[i]).replace('\xa0', ' '))]))
    bookstar = []
    for i in books:
        bookstar.append(i.text)
    peoples
    return [bookno, bookname, bookinfo, tro1, tro2, int(bookp.replace('人评价', '')), ', '.join(bookstar), bookalot, bookamulu, '<hr />'.join(peoples)]

def booktag(url_content, path = 'web/booktag.xlsx'):
    """
    抓取标签提取 写入 Excel

    """
    soup = BeautifulSoup(url_content, 'html.parser') # 开始解析
    booktag1 = soup.select('div#content div.article div div')
    # print(booktag1[0])
    taglist = [['标签类别', '标签名', '链接']]
    for booktag2 in booktag1:
        soup1 = BeautifulSoup(str(booktag2), 'html.parser') # 开始解析
        booktag2 = soup1.find('a', attrs={'class': 'tag-title-wrapper'})
        type = booktag2['name'] # 标签类别
        booktag3 = soup1.findAll('a', attrs={'class': 'tag'})
        for i in booktag3:
            tag = i.string # 标签名
            taglink = i['href'] # 链接
            taglist.append([type, tag, taglink])
    print(taglist)
    writeexcel(path, taglist)
    print("写入 EXCEL 成功")

def testbooktag():
    file = open('web/booktag.html', 'rb')
    content = file.read()
    booktag(content)

def testmanybook():
    file = open('web/books.html', 'rb')
    content = file.read()

```

```

books = manybook(content)
for i in books:
    print(i)

def testonebook():
    file = open('web/book.html','rb')
    content = file.read()
    book = onebook(content)
    for i in book:
        print(i)
        print('*'*50)

if __name__=='__main__':
    # testmanybook()
    testonebook()
    #testbooktag()

```

图片抓取程序

因为时间因素没有抓，请仿照抓取

```

# -*- coding:utf-8 -*-
import os, urllib.request

# 保存图书封面
# 根据文件名创建文件
def createFileWithFileName(localPathParam,fileName):
    totalPath=localPathParam+'/'+fileName+'.jpg'
    if not os.path.exists(totalPath):
        file=open(totalPath,'a+')
        file.close()
        return totalPath

# 根据图片的地址，下载图片并保存在本地
def getAndSaveImg(imgUrl, fileName, localPath='../image'):
    if(len(imgUrl)!= 0):
        urllib.request.urlretrieve(imgUrl,createFileWithFileName(localPath,fileName))

```


抓取辅助工具

gethtml.py

```
# -*- coding:utf-8 -*-
import urllib.request
import urllib.parse
import urllib.request, urllib.parse, http.cookiejar
from bs4 import BeautifulSoup
__author__ = 'hunterhug'

def getHtml(url):
    """
    伪装头部并得到网页内容

    """
    cj = http.cookiejar.CookieJar()
    opener =
urllib.request.build_opener(urllib.request.HTTPCookieProcessor(cj))
    useragent3 = 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101
Firefox/38.0'

Accept='text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'
    cookie = ""'"Cbid="FccuPZmq//0";
viewed="25923455_25867785_1520363_6397086_6431094_5338398_25862578_2658901
8_1002898_5922149"; gr_user_id=9fee2430-40d3-4f9d-a5aa-d3e295531497;
__utma=30149280.372796509.1449028995.1450149245.1450165592.16;
__utmz=30149280.1450149245.15.12.utmcsr=baidu|utmccn=(organic)|utmcmd=orga
nic; __utma=81379588.1458595321.1449028995.1450149245.1450165592.11;
__utmz=81379588.1450149245.10.10.utmcsr=douban.com|utmccn=(referral)|utmcm
d=referral|utmcct=/;
__pk_id.100001.3ac3=ff3d4ecea493334f.1449028995.11.1450166087.1450149661.;
ll="118281";
__pk_ref.100001.3ac3=%5B%22%22%2C%22%22%2C1450165591%2C%22http%3A%2F%2Fwww.
douban.com%2F%22%5D; ap=1; ct=y; ps=y; __utmv=30149280.13418;
ue="569929309@qq.com"; push_noty_num=0; push_doumail_num=0; __utmc=30149280;
__utmc=81379588; gr_session_id=8951b2b8-06a3-4a07-b066-0ee33d2be006;
__pk_ses.100001.3ac3=*; __utmb=30149280.2.10.1450165592; __utmt_douban=1;
__utmb=81379588.2.10.1450165592; __utmt=1""
    opener.addheaders = [('User-Agent',useragent3),
                          ('Accept',Accept),
                          ('Cookie', cookie)]
```

```

urllib.request.install_opener(opener)

html_bytes = urllib.request.urlopen(url).read()
html_string = html_bytes.decode('utf-8', 'ignore')
return html_string

def getBinaryHtml(url,daili='42.96.162.252:3128'):
    """
    伪装头部并得到网页原始内容

    """
    cj = http.cookiejar.CookieJar()
    # 设置IP 代理
    # http://www.youdaili.net/
    # http://www.youdaili.net/Daili/http/3917.html
    proxy_support = urllib.request.ProxyHandler({'http':'http://'+daili})
    # 开启代理支持
    opener = urllib.request.build_opener(proxy_support,
urllib.request.HTTPCookieProcessor(cj), urllib.request.HTTPHandler)

    #opener =
urllib.request.build_opener(urllib.request.HTTPCookieProcessor(cj))
    # 用户代理http://blog.csdn.net/lvjin110/article/details/12944397
    useragent = 'Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+
(KHTML, like Gecko) Version/3.0 Mobile/1C28 Safari/419.3'
    useragent1 = 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/41.0.2272.101 Safari/537.36'
    useragent2 = 'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)'
    useragent3 = 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101
Firefox/38.0'

Accept='text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'
    opener.addheaders = [('User-Agent',useragent3),
                        ('Accept',Accept),
                        ('Cookie', '4564564564564564565646540')]

    urllib.request.install_opener(opener)

    html_bytes = urllib.request.urlopen(url).read()
    return html_bytes

def getSoup(html_content,parse='html.parser'):

```

```

"""
    得到网页解析后的对象，方便分拆数据

"""
    return BeautifulSoup(html_content, parse)

def test():

getBinaryHtml('http://www.douban.com/tag/%E5%B0%8F%E8%AF%B4/book?start=195')

if __name__ == '__main__':
    tag = getBinaryHtml('http://book.douban.com/tag/')
    file = open('../web/booktag.html', 'wb')
    file.write(tag)
    file.close()
    # content1 = getHtml("http://www.douban.com/tag/%E5%B0%8F%E8%AF%B4/book")
    # file1 = open('../web/books.html', 'wb')
    # content2 =
getHtml("http://book.douban.com/subject/25862578/?from=tag_all")
    # file2 = open('../web/book.html', 'wb')
    # file1.write(content.encode('utf-8'))
    # file2.write(content.encode('utf-8'))
    # file1.close()
    # file2.close()
    # test()

```

文件处理工具

filetool.py

```

__author__ = 'hunterhug'
import os.path
from openpyxl import Workbook
from openpyxl import load_workbook
import re
# 找出文件夹下所有html 后缀的文件
def listfiles(rootdir, prefix='.html'):
    file = []
    for parent, dirnames, filenames in os.walk(rootdir):
        if parent == rootdir:
            for filename in filenames:

```

```

        if filename.endswith(prefix):
            file.append(filename)
        return file
    else:
        pass

def rlistfiles(rootdir, prefix='.html'):
    file = []
    for parent, dirnames, filenames in os.walk(rootdir):
        for filename in filenames:
            if filename.endswith(prefix):
                #file.append(filename)
                print(str(parent)+'/'+filename)
            else:
                pass
    return file

def writeexcel(path, content, sheetname='标签抓取表'):
    wb=Workbook()
    sheet=wb.create_sheet(0,sheetname)
    row = 1
    col = 1
    for i in content:
        for j in i:
            sheet.cell(row=row,column=col).value = j
            col = col + 1
        col = 1
        row =row + 1
    wb.save(path)
    print("保存数据成功! ")

def readexcel(path):
    excelcontent = []
    wb2=load_workbook(path)
    sheetnames = wb2.get_sheet_names()
    ws=wb2.get_sheet_by_name(sheetnames[0])
    row=ws.get_highest_row()
    col=ws.get_highest_column()
    # print("列数: ",ws.get_highest_column())
    # print("行数: ",ws.get_highest_row())

    for i in range(0,row):
        rowcontent = []
        for j in range(0,col):

```

```

        if ws.rows[i][j].value:
            rowcontent.append(ws.rows[i][j].value)
        excelcontent.append(rowcontent)
    print("读取数据成功! ")
    return excelcontent

# 去除标题中的非法字符 (Windows)
def validateTitle(title):
    rstr = r"[\/\:\*\?\\"<\>\|]" # '/\:*? ">/'
    new_title = re.sub(rstr, "", title)
    return new_title

if __name__ == '__main__':
    rlistfiles('../book')

```

数据库辅助工具

mysql.py

```

# -*- coding:utf-8 -*-
import pymysql

class Mysql:
    """
    对pymysql 的简单封装, 实现基本的连接
    """

    def __init__(self, host, user, pwd, db):
        self.host = host
        self.user = user
        self.pwd = pwd
        self.db = db
        self.cur=self.__GetConnect()

    def __GetConnect(self):
        """
        得到连接信息
        返回: conn.cursor()
        """

        if not self.db:
            raise (NameError, "没有设置数据库信息")
        self.conn = pymysql.connect(host=self.host, user=self.user,

```

```

passwd=self.pwd, db=self.db, charset="utf8")
    cur = self.conn.cursor()
    if not cur:
        raise (NameError, "连接数据库失败")
    else:
        return cur

def ExecQuery(self, sql):
    """
    执行查询语句
    返回的是一个包含tuple的list, list的元素是记录行, tuple的元素是每行记录的
    字段

    调用示例:
        ms =
MYSQL(host="localhost", user="sa", pwd="123456", db="PythonWeiboStatistics")
        resList = ms.ExecQuery("SELECT id, NickName FROM WeiBoUser")
        for (id, NickName) in resList:
            print str(id), NickName
    """
    self.cur.execute(sql)
    #print("查询语句: "+sql)
    resList = self.cur.fetchall()
    return resList

def ExecNonQuery(self, sql):
    """
    执行非查询语句

    调用示例:
        cur = self.__GetConnect()
        cur.execute(sql)
        self.conn.commit()
        self.conn.close()
    """
    try:
        self.cur.execute(sql)
        self.conn.commit()
        print('执行语句成功')
    except Exception: # 出现异常回滚
        self.conn.rollback()
        print('执行 SQL 语句失败: '+sql)
        raise

```

```

def __del__(self):
    self.cur.close()

def init():
    return Mysql(host="localhost", user="root", pwd="6833066",
db="doubanbook")

def testinsert():
    mysql1 = init()
    mysql1.ExecNonQuery("insert into `bookdetial` (booknafme) values ('你哈'
)")

def testselect():
    mysql1 = init()
    print(mysql1.ExecQuery('SELECT bookname,bookkind,bookno FROM booktag
group by bookno limit 0,5;')[0][0])
    print(mysql1.ExecQuery('SELECT bookname,bookkind,bookno FROM booktag
group by bookno limit 5,5;'))
    print('-'*50)
    print(mysql1.ExecQuery('SELECT bookname,bookkind,bookno FROM booktag
group by bookno limit 0,10;'))

def initdoubanbook():
    mysql = pymysql.connect(host="localhost", user="root", passwd="6833066",
charset="utf8")
    cur = mysql.cursor()
    createsql = """
CREATE SCHEMA `doubanbook` ;
use `doubanbook`;
CREATE TABLE `book` (
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增 ID',
  `bookname` varchar(100) NOT NULL COMMENT '书名',
  `bookurl` varchar(150) NOT NULL COMMENT '书入口',
  `booking` varchar(150) DEFAULT NULL COMMENT '书图片',
  `bookinfo` varchar(250) DEFAULT NULL COMMENT '书出版信息',
  `bookstar` varchar(45) DEFAULT NULL COMMENT '书评价星数',
  `bookno` varchar(45) NOT NULL COMMENT '书编号',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='书表';

CREATE TABLE `booktag` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `bookname` varchar(100) DEFAULT NULL COMMENT '书名',

```



```

`bookno` varchar(45) DEFAULT NULL COMMENT '书编号',
`booktag` varchar(45) DEFAULT NULL COMMENT '书标签',
`bookkind` varchar(45) DEFAULT NULL COMMENT '书分类',
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='书标签';""
try:
    cur.execute(createsql)
    mysql.commit()
    return createsql
except:
    mysql.rollback()
    print("执行失败")

if __name__ == '__main__':
    # testinsert()
    testselect()

```

代理 IP 存放工具

daili.py

```

__author__ = 'hunterhug'
import random
import re
# 代理ip 函数
# 183.239.167.122:8080
def daili():
    geshi=re.compile(r'(.*)@(.*)')
    file =open('daili.txt','rb')
    data=file.read().decode('utf-8','ignore').split('\n')
    location=[]
    #random.shuffle(data) # ip 数组打乱
    for i in range(0,len(data)):
        temp=geshi.match(data[i]).group(1).split('.')
        location.append(geshi.match(data[i]).group(2))
        data[i]='.'.join([temp[1],temp[2],temp[3]])
    file.close()
    file =open('daili1.txt','w')
    file.write('\n'.join(data))
    file.close()
    return data

if __name__=='__main__':

```

```
a,b=daili()  
print(a)
```

代理文件格式

daili.txt

网址: <http://www.youdaili.net/>

```
1.179.176.37:8080@HTTP#泰国 TOT 公共有限公司  
1.179.146.153:8080@HTTP#泰国 TOT 公共有限公司  
1.179.198.37:8080@HTTP#泰国 TOT 公共有限公司  
1.234.45.50:3128@HTTP#韩国 SK 电讯  
1.255.53.81:80@HTTP#韩国 SK 电讯  
5.22.195.148:80@HTTP#【匿】伊朗  
5.135.161.61:3128@HTTP#法国  
5.141.9.86:8080@HTTP#俄罗斯  
5.160.247.16:8080@HTTP#伊朗  
5.196.99.243:3128@HTTP#德国  
5.196.208.4:3128@HTTP#【匿】德国  
23.24.89.193:7004@HTTP#【匿】美国 新泽西州月桂山镇 Comcast 商业通信有限公司  
23.97.213.142:8118@HTTP#【匿】美国 Microsoft 公司  
27.254.47.203:80@HTTP#泰国
```

数据库设计

```
CREATE TABLE `book` (  
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增 ID',  
  `bookname` varchar(100) NOT NULL COMMENT '书名',  
  `bookurl` varchar(150) NOT NULL COMMENT '书入口',  
  `bookimg` varchar(150) DEFAULT NULL COMMENT '书图片',  
  `bookinfo` varchar(250) DEFAULT NULL COMMENT '书出版信息',  
  `bookstar` varchar(45) DEFAULT NULL COMMENT '书评价星数',  
  `bookno` varchar(45) NOT NULL COMMENT '书编号',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='书表';  
  
CREATE TABLE `booktag` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `bookname` varchar(100) DEFAULT NULL COMMENT '书名',  
  `bookno` varchar(45) DEFAULT NULL COMMENT '书编号',  
  `booktag` varchar(45) DEFAULT NULL COMMENT '书标签',  
  `bookkind` varchar(45) DEFAULT NULL COMMENT '书分类',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='书标签';
```

```

CREATE TABLE `bookdetial` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `bookname` varchar(100) NOT NULL COMMENT '书名',
  `bookno` varchar(45) NOT NULL COMMENT '书编号',
  `bookinfo` text COMMENT '书出版信息',
  `bookintro` text COMMENT '书介绍',
  `authorintro` text COMMENT '作者介绍',
  `peoples` int(11) DEFAULT NULL COMMENT '评价人数',
  `starts` varchar(100) DEFAULT NULL COMMENT '星级情况',
  `other` text COMMENT '其他信息',
  `mulu` mediumtext COMMENT '图书目录',
  `comments` mediumtext COMMENT '评论人',
  PRIMARY KEY (`id`),
  UNIQUE KEY `bookno_UNIQUE` (`bookno`)
) ENGINE=InnoDB AUTO_INCREMENT=18149 DEFAULT CHARSET=utf8 COMMENT='图书详情表';

```

四、总结

通过网络编程，实现了抓取豆瓣图书列表，提取其关键内容，解决了之前复制粘贴效率低的问题。收集的此些数据可以用来做很多事，同理，根据这种思路，可以抓取网络上很多有用的数据。

由于抓取时间有限，目前图书封面和图书评论还待抓，部分程序需要修正，以及反爬虫的问题还有待讨论。以下为程序源码及产物说明：

```

1 本爬虫程序目录如下：
2  ----book 抓取的图书详情页
3      ----文学      大分类
4          ----1000121昆虫记.html 标号+标题
5      ----文化
6      ----生活
7      ----流行
8      ----经管
9  ----books 提取的图书列表页
10     ----文学      大分类
11         ----茨威格.xlsx 标签
12     ----文化
13     ----生活
14     ----流行
15     ----经管
16  ----data 提取的数据库文件
17     ----doubanbook.book.sql 图书基本信息
18     ----doubanbook.booktag.sql 图书标签信息
19  ----image 抓取的图片
20  ----web 抓取的图书列表页
21     ----文学      大分类
22         ----茨威格 标签
23             ----0.html 列表页
24             ----1.html
25     ----文化
26     ----生活
27     ----流行
28     ----经管
29     ----book.html 测试的图书详情页
30     ----books.html 测试的图书列表页
31     ----booktag.html 测试图书标签页
32     ----booktag.xlsx 提取的图书标签页
33
34  ----tool 抓取工具
35  ----源码
36  ----pack 打包的所有东西，安装上面目录解压

```

参考

百度百科、谷歌、维基、各大 IT 社区

博客：<http://www.cnblogs.com/nima/p/4989482.html>

Python：<https://www.python.org/>

Mysql：<http://www.mysql.com/>

代码附件：见光盘